Supplementary materials

1. Calculation of ITR

Assuming that the 'timeout' commands are equivalent to 'no decision', we calculate ITR as in equation 1. This measure accounts for imbalance in the dataset and bias in the decisions [1].

$$ITR = \sum_{j=1}^{N_c+1} \sum_{i=1}^{N_c} p(x_i)p(y_j|x_i)log_2(p(y_j|x_i)) - \sum_{j=1}^{N_c+1} p(y_j)log_2(p(y_j)), \tag{1}$$

where

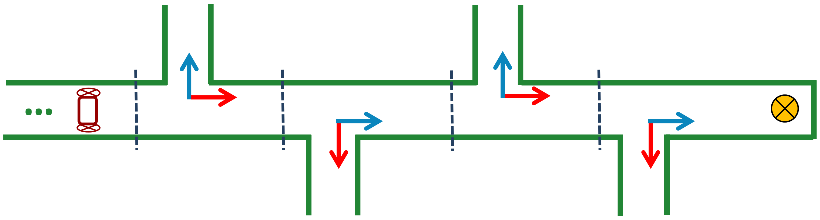
$$p(y_j) = \sum_{i=1}^{N_c} p(x_i)p(y_j|x_i), \tag{2}$$

where $p(y_j|x_i)$ corresponds to the (i, j) element in the normalised confusion matrix $(CM[i, j]/N_i)$, N_c is the number of classes, and $p(x_i)$ is the prior for each class. In order to obtain the bit rate per minute, ITR should be multiplied by the number of trials per minute. The inter-trial interval (ITI) in the experiment was variable (6 s to 7 s), however, it was considered as 6 s for the analysis.

2. Simulated online adaptive assistance in a robotic scenario

As mentioned before, estimating the subject's performance in terms of short or long command delivery can be beneficial in some applications, e.g., MI-based navigation of a telepresence robot. We use the data in the MI-BCI game to simulate such an application. An example scenario is depicted in Supp-Fig. 1. The robot needs to reach the yellow target as fast as possible passing several junctions on the way. At each junction, the subject needs to deliver a command, either to the right (red) or to the left (blue). The junctions can be observed by the subject only when the robot reaches the dashed lines before them. At the moment, she/he has a certain amount of time to deliver a command. If the user does not deliver a command on time, the robot stops at the junction, waiting for the next command. In case of a wrong command, the robot turns to the wrong direction and the user needs to deliver two additional correct commands to compensate for the wrong one.

Assuming that the user needs to deliver n correct commands to reach the target (i.e., n junctions), we simulate the same conditions as in the adaptive assistance for the BCI game: *fixed timeout* and *adaptive assistance* (cf., Section 2.1.3). Here, the fixed timeout condition corresponds to a case in which the robot moves with a constant speed; since the subject starts delivering a command at a certain point before the junction (the dashed line in Supp-Fig. 1), she/he always has the same amount of time (i.e., the timeout) to deliver a command. In the adaptive assistance case, the robot slows down if the performance estimator predicts a long CDT so that the subject has 8 s to deliver a command. Otherwise, the default speed is maintained. We performed two types of simulation. In the first case, we use the results from the Evaluation sessions of the BCI game. In the second case, we use the results of those sessions where adaptive assistance was provided (Figure 3).



Supp-Fig. 1 – A robotic navigation scenario, in which the users are required to reach the yellow target as fast as possible. They can see the junctions when they reach the dashed line, at which point they should deliver a command either to the right (red) or to the left (blue). In case of delivering a wrong command, the robot turns into the sides and the user needs to deliver two additional correct commands to compensate for it. In case of not delivering a command on time, the robot stops at the junction, waiting for the next command.

2.1. Simulation based on performance estimation in the 'Evaluation' sessions

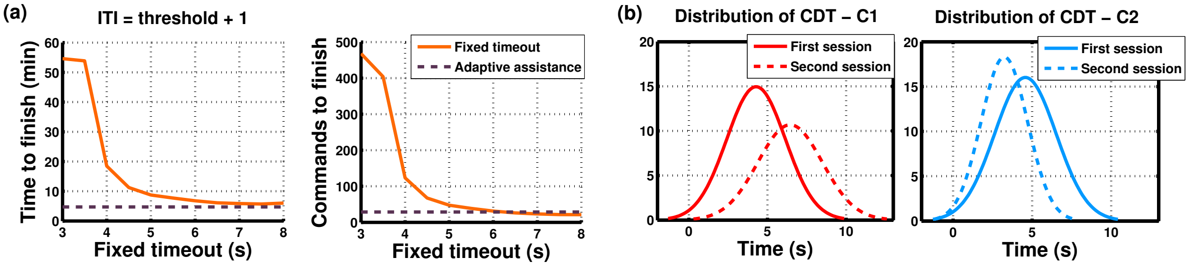
For each condition, a sequence of trials (right or left commands) were randomly selected from the recordings in the Evaluation sessions of the MI-BCI game (Figure 3) to simulate the sequence of decisions at the junctions. Based on the performance of these sessions, the time to complete the task and the number of commands required to finish were assessed for both conditions. For the simulation, it was assumed that there were n = 10 junctions on the way to the target. This number was chosen to allow non-repetitive random selection of the trials. The inter-trial interval (ITI) was assumed to change based on the speed of the robot. That is, the higher the fixed timeout, the slower the robot and therefore, the higher the ITI. ITI was chosen as $\textit{fixed timeout} + 1$ s in the simulations. The analysis was repeated 100 times using different random selections of trials.

For the first condition (fixed timeout), different timeout values were tested (from 3 to 8, with a step size of 0.5). For the second condition (adaptive assistance), the results of the performance estimator (based on the outcome of the performance estimator in single trials of the Evaluation session) were used to define when to slow down the robot.

2.2. Simulation based on online adaptive assistance results

The simulated online results should be interpreted carefully, as the user may change strategies according to the system performance and feedback [1]. In order to be closer to the case of closed-loop navigation task with adaptive assistance, we simulated the robotic application using the results of the game where the level of assistance is regulated based on the users' performance (Section 2.1.3). In this case, the user receives feedback on the performance of the BCI (i.e., movement of the platform), as well as the provided assistance (i.e., slowing down of the parachutist if long CDTs are predicted).

The results achieved in the adaptive assistance sessions (command delivery results, i.e., hit, miss, or timeout) of the game were used for the simulation. In both conditions, $n = 5$



Supp-Fig. 2 – *Simulated robotic scenario using online MI-BCI game sessions*: **(a)** The time and number of commands required to finish the navigation task (10 junctions) comparing a fixed time to the adaptive assistance case. **(b)** The distribution of CDT over two sessions for sb8 shows rather high variations.

junctions were considered on the way to the target. As the number of trials is limited in this case, considering $n = 10$ may result in repetitive selection of trials. ITI was considered to be $t_{sl} + 1$ in both cases. The analysis was performed 100 times.

2.3. Results

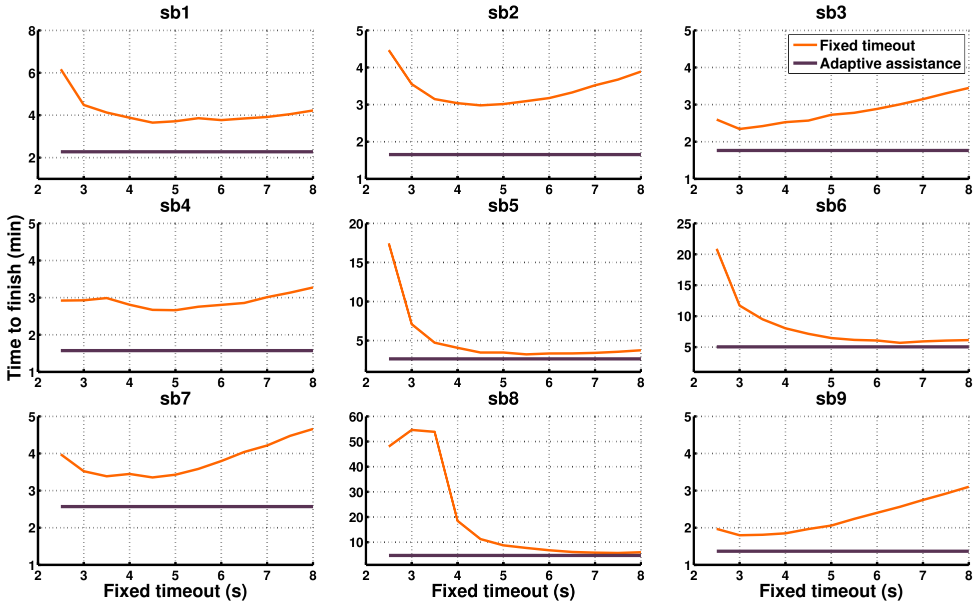
2.3.1. Simulation based on performance estimation in the 'Evaluation' sessions

The simulated results of implementing adaptive assistance in a robotic scenario are illustrated in Supp-Fig. 2 for one subject (sb8). We detail the results on this subject, as she showed rather high performance variations across experimental sessions (Evaluation sessions of the game). These results reveal that when a fixed timeout is used, the user would typically require longer time than the adaptive case to finish the navigation task unless long thresholds are chosen. The same pattern is observed for the number of commands to accomplish the task. Nevertheless, the choice of a fixed timeout for a command delivery may be more challenging in case of high variations across sessions. For example, for this subject, we may select a fixed timeout of 7 s based on the CDT distribution of C1 in the first session for the majority of commands to be delivered on time. Yet, this threshold will result in failure in at least half of the trials in the second session (Supp-Fig. 2.b).

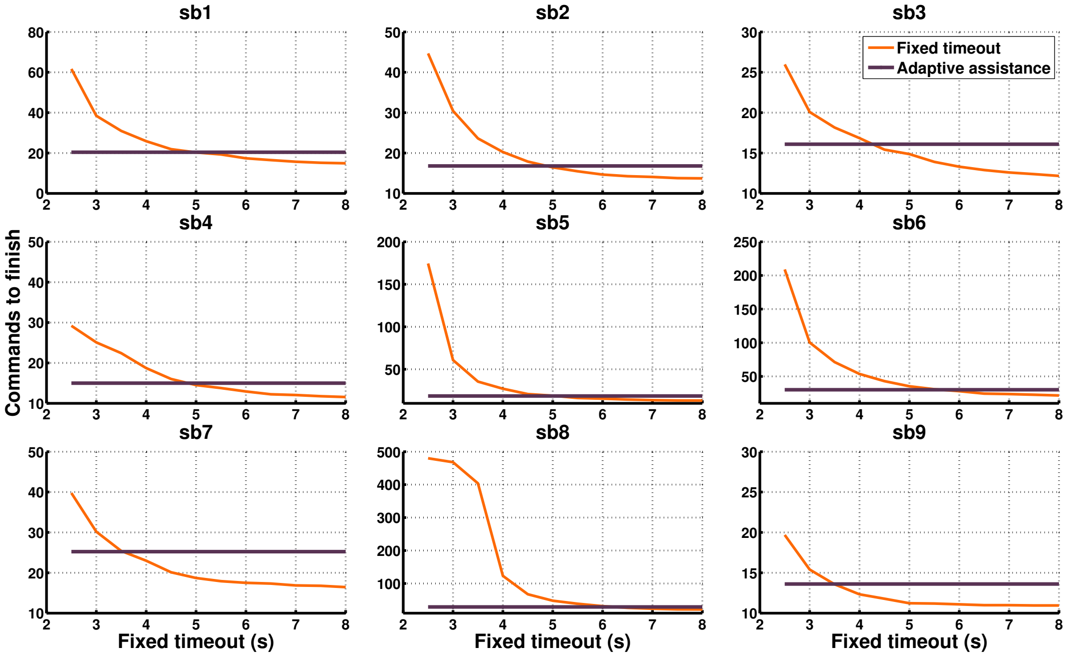
This simulated comparison for all subjects is shown in Supp-Fig. 3 and 4. According to the results, the time to finish the task is always higher when using a fixed timeout than when using the adaptive assistance. By increasing the threshold, the number of commands to finish the task decreases often reaching a minimum value.

2.3.2. Simulation based on online adaptive assistance results

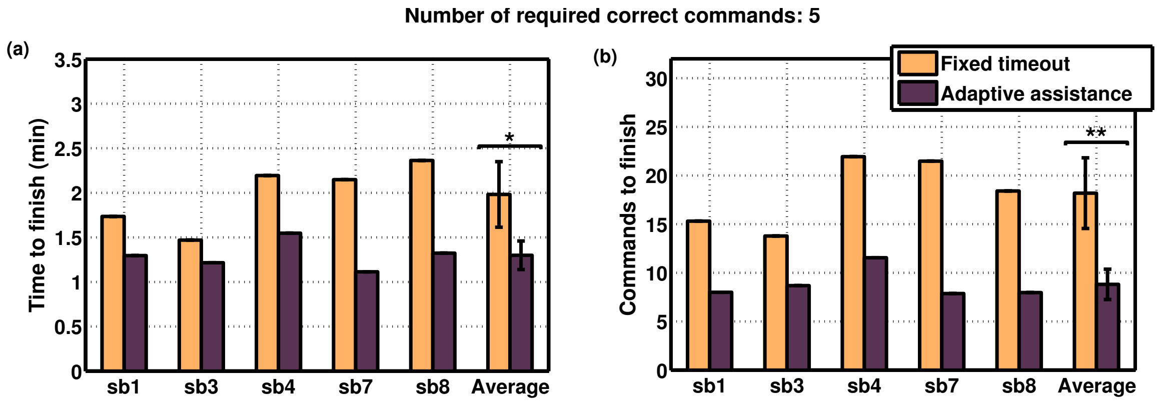
The advantages of providing adaptive assistance are supported by these results as well. As illustrated in Supp-Fig. 5, the number of commands and the time required to finish the navigation task (with 5 junctions on the way to the target) reduces significantly when the adaptive assistance is provided (time $p < 0.05$ and commands $p < 0.01$, respectively). This



Supp-Fig. 3 – *Simulated robotic scenario using online MI-BCI game sessions*: The time required to finish the navigation task when there are 10 junctions on the way to the target comparing adaptive assistance and selecting a fixed time for command delivery.



Supp-Fig. 4 – *Simulated robotic scenario using online MI-BCI game sessions*: Number of commands required to finish the navigation task when there are 10 junctions on the way to the target comparing adaptive assistance and selecting a fixed time for command delivery.



Supp-Fig. 5 – *Simulated robotic scenario using online MI-BCI game sessions*: The time and the number of commands required to finish the navigation task with 5 junctions on the way to the target. Providing adaptive assistance significantly reduces the time and number of commands required comparing to having always a fixed timeout for a command to be delivered ($** \ p < 0.01, * \ p < 0.05$; Wilcoxon rank sum test).

demonstrates the clear benefit of providing adaptive assistance.

References

[1] E. Thomas, M. Dyson, and M. Clerc, "An analysis of performance evaluation for motor-imagery based BCI," *Journal of Neural Engineering*, vol. 10, no. 3, p. 031001, 2013.